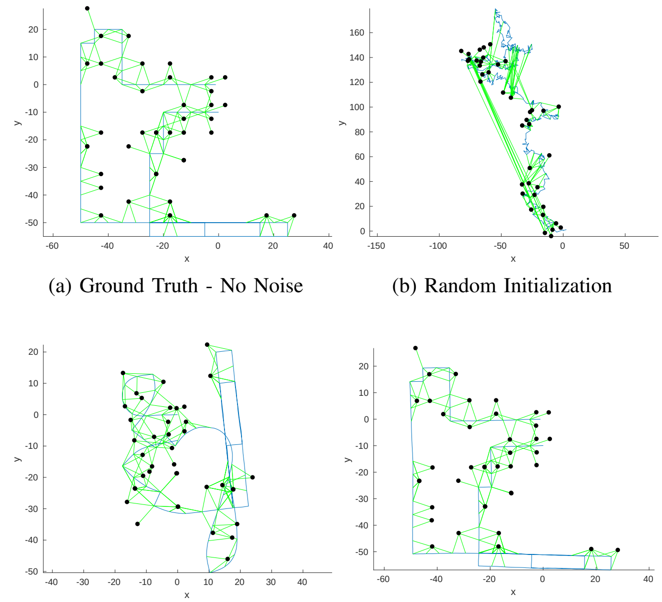
Guaranteed Globally Optimal Planar Pose Graph and Landmark SLAM via Sparse-Bounded Sums-of-Squares Programming

Joshua G. Mangelson, Jinsun Liu, Ryan M. Eustice, and Ram Vasudevan

***Abstract*— Autonomous navigation requires an accurate model or map of the environment. While dramatic progress in the prior two decades has enabled large-scale simultaneous localization and mapping (SLAM), the majority of existing methods rely on non-linear optimization techniques to find the maximum likelihood estimate (MLE) of the robot trajectory and surrounding environment. These methods are prone to local minima and are thus sensitive to initialization. Several recent papers have developed optimization algorithms for the Pose-Graph SLAM problem that can certify the optimality of a computed solution. Though this does not guarantee *a priori* that this approach generates an optimal solution, a recent extension has shown that when the noise lies within a critical threshold that the solution to the optimization algorithm is guaranteed to be optimal. To address the limitations of existing approaches, this paper illustrates that the Pose-Graph SLAM and Landmark SLAM can be formulated as polynomial optimization programs that are sum-of-squares (SOS) convex. This paper then describes how the Pose-Graph and Landmark SLAM problems can be solved to a global minimum without initialization regardless of noise level using the sparse bounded degree sum-of-squares (Sparse-BSOS) optimization method. Finally, the superior performance of the proposed approach when compared to existing SLAM methods is illustrated on graphs with several hundred nodes.**



(a) Ground Truth - No Noise (b) Random Initialization

(c) Levenberg-Marquardt (d) Global Optimal - SBSOS-SLAM

Fig. 1: Estimated Landmark SLAM solution for the first 430 nodes(40 landmarks) of the CityTrees10000 dataset [8]. (a) shows the groundtruth pose and landmark positions before noise is added. Levenberg-Marquardt was randomly initialized and becomes trapped in a local minimum. (b) shows the random initialization and (c) shows the LM solution. (d) shows the globally optimal solution found by SBSOS-SLAM. Our algorithm formulates the Pose Graph and Landmark SLAM problems as SOS-convex optimization problems and is guaranteed to find the globally optimal solution without any initialization.

I. Introduction

An accurate map of the environment is essential for safe autonomous navigation in the real-world [1]. An error in the map has the potential to cause loss of life in self-driving car applications or the loss of millions of dollars of assets when performing underwater or space exploration tasks. Despite the importance of accurate mapping, the majority of algorithms used for simultaneous localization and mapping (SLAM) are prone to local minima and are sensitive to initialization. Troublingly, verification of these maps is either performed by visual inspection or not at all.

There has been significant recent interest in developing optimization and estimation algorithms that provide mathematical guarantees on whether a computed solution is or is close to the global optimum and is therefore true maximum *a posteriori* (MAP) estimate of the map [2–7]. These algorithms either use a relaxation or the dual of the original problem to find a solution. As a result these methods either return an approximate solution, are only able to certify the optimally of a solution after it has been computed, or are only able to return the global solution if the graph meets certain requirements related to limits on noise measurement. In addition, with the exception of [6], these methods are focused on pose-graph optimization and are unable to handle landmark position measurements or are unable to estimate landmark positions.

As depicted in Figure 1d, the contributions of this paper are the following:

1) We formulate the pose graph and landmark planar SLAM as polynomial optimization programs.
2) We describe how the sparse bounded degree sum-of-squares (Sparse-BSOS) hierarchy of semidefinite programs (SDP) can be used to find its solution [9, 10].
3) We show that both the pose-graph and landmark SLAM problems are SOS convex meaning that they have a single globally optimal solution that can be found exactly by solving the first step of the Sparse-BSOS hierarchy.

*This work was supported by the Office of Naval Research under awards N00014-16-1-2102 and N00014-18-1-2575.

J. Mangelson, J. Liu, R. Eustice, and R. Vasudevan are at the University of Michigan, Ann Arbor, MI 48109, USA. {mangelso, jinsunl, eustice, ramv}@umich.edu.

II. Related Work

SLAM refers to the problem of estimating the trajectory of a robotic vehicle over time while simultaneously estimating a model of the surrounding environment [1]. Initial algorithms used extended Kalman filter and particle filter based methods to simultaneously estimate the position of the robot and the position of observed landmarks in the environment [11–13], which we refer to as the Landmark SLAM problem. Since these methods had challenges scaling to larger datasets, researchers began applying information filter and maximum likelihood estimate (MLE) based methods which could exploit sparsity to solve larger instances of the SLAM problem. To improve the sparsity of the problem, research shifted to solving the Pose Graph SLAM problem wherein the landmarks are marginalized out and only the pose of the robot is optimized over at each time step. The majority of modern SLAM algorithms seek to find the MLE of the robot trajectory through the use of nonlinear estimation based techniques [8, 14–16]. However, the nonlinear optimization algorithms used in these methods are dependent on initialization.

Several algorithms leverage theory from the field of convex optimization to overcome this dependence on initialization [17]. Optimization over the special euclidean group ($\mathrm{SE}(d)$) has generally been considered a non-convex problem and thus the majority of algorithms rely on some form of convex relaxation to estimate an approximate and sometimes exact solution to the problem. For instance the Pose Graph and Landmark SLAM problems have been formulated as a non-convex quadratically constrained quadratic program, which was then relaxed into an SDP [6, 7]. Rosen et al. [4] relaxes optimization over the special orthogonal group ($\mathrm{SO}(d)$) to the convex hull of $\mathrm{SO}(d)$ which can be represented using convex semidefinite constraints. Since each of these methods only provide an approximate solution to the SLAM problem, they are usually only used as an initial stage and their output is then used to initialize a non-linear optimization method [8, 16].

A number of methods take advantage of Lagrangian Duality to convert the Pose Graph SLAM problem into a convex optimization problem that is equivalent to the original optimization problem if the duality gap is zero [17, Section 5.3.2]. Carlone et al. [5] uses Lagrangian Duality to develop a pair of methods to verify if a computed solution is globally optimal. Carlone et al. [3] applies a similar technique to the planar Pose Graph SLAM problem. SE-Sync proposed by Rosen et al. [2] extends this prior work and dramatically increases the scalability of the algorithm by taking advantage of a technique called the Riemannian staircase [18] that enables efficient optimization over semidefinite matrices if the solution has low-rank. These methods are only guaranteed to find the globally optimal solution if the measurement noise in the problem lies below a critical threshold and are restricted to the case of Pose Graph SLAM where factors are relative pose measurements in $\mathrm{SE}(d)$.

This paper shows that the Landmark and Pose Graph SLAM problems can be formulated as SOS convex polynomial optimization problems which can be optimized to the global minimum (without initialization) through the use of Sparse-BSOS optimization.

The convexity of this formulation also allows us to relax the requirement of limited measurement noise and extend the formulation to a broader class of SLAM problems.

III. Polynomial Optimization SLAM Formulation

This section formulates the Pose Graph and Landmark SLAM problems as polynomial optimization programs.

A. Polynomial Optimization

A polynomial optimization program is an optimization problem of the following form [9, Section 2.2]:

$$f^* = \min_{\mathbf{x}} \{f(\mathbf{x}) : \mathbf{x} \in \mathbf{K}\} \quad (1)$$

where $f \in \mathbb{R}[\mathbf{x}]$, $\mathbb{R}[\mathbf{x}]$ is the ring of all possible polynomials in the variable $\mathbf{x} = (x_1, \ldots, x_N)$, and $\mathbf{K} \subset \mathbb{R}^N$ is the semi-algebraic set

$$\mathbf{K} = \{\mathbf{x} \in \mathbb{R}^N : 0 \le g_j(\mathbf{x}) \le 1, j = 1, \ldots, M\}, \quad (2)$$

for polynomials $g_j \in \mathbb{R}[\mathbf{x}], j = 1, \ldots, M$.

B. Pose Graph SLAM

In planar Pose Graph SLAM, one estimates the pose of the robot, $(\mathbf{R}_i, \mathbf{t}_i) \in \mathrm{SE}(2)$ with respect to a static global reference frame at each time steps $i \in \{1, \ldots, n\}$, by minimizing the error in a set of m_{Rel} relative pose measurements $(\bar{\mathbf{R}}_{ij}, \bar{\mathbf{t}}_{ij}) \in \mathrm{SE}(2)$. The set of available measurements can be represented by the set of edges, $E = \{i_k, j_k\}_{k=1}^{m_{Rel}} \subset \{1, \ldots, n\} \times \{1, \ldots, n\}$, in the corresponding factor graph. We denote the pose of the robot at time step i by the matrix $\mathbf{H}_i = [\mathbf{R}_i | \mathbf{t}_i]$ and the relative pose measurement that relates the pose of the robot at time steps i and j by $\bar{\mathbf{H}}_{ij} = [\bar{\mathbf{R}}_{ij} | \bar{\mathbf{t}}_{ij}]$. We assume that each $\bar{\mathbf{R}}_{ij}$ and $\bar{\mathbf{t}}_{ij}$ are conditionally independent given the true state, that $\bar{\mathbf{R}}_{ij} \sim \text{Langevin}(\mathbf{R}_{ij}, \omega^2_{\mathbf{R}_{ij}})$, and that $\bar{\mathbf{t}}_{ij} \sim \mathcal{N}(\mathbf{t}_{ij}, \Omega^{-1}_{\mathbf{t}_{ij}})$, where $(\mathbf{R}_{ij}, \mathbf{t}_{ij})$ is the true relative pose, $\omega^2_{\mathbf{R}_{ij}}$ is the concentration parameter of the Langevin Distribution, and $\Omega_{\mathbf{t}_{ij}} = \text{blkdiag}(\omega^2_{x_{ij}}, \omega^2_{y_{ij}})$ is the information matrix of $\bar{\mathbf{t}}_{ij}$. Note $\text{blkdiag}(\omega^2_{x_{ij}}, \omega^2_{y_{ij}})$ denotes a block diagonal matrix whose diagonal elements are equal to $\omega^2_{x_{ij}}$ and $\omega^2_{y_{ij}}$.

Under these assumptions, the MLE solution to the planar pose graph SLAM problem is equivalent to:

$$\underset{\mathbf{H}_1, \ldots, \mathbf{H}_n \in \mathrm{SE}(2)}{\operatorname{argmin}} \sum_{(i,j) \in E} \omega^2_{\mathbf{R}_{ij}} \left|\left|\mathbf{R}_j - \mathbf{R}_i \bar{\mathbf{R}}_{ij}\right|\right|^2_F + ||\mathbf{t}_j - \mathbf{t}_i - \mathbf{R}_i \bar{\mathbf{t}}_{ij}||^2_{\Omega_{\mathbf{t}_{ij}}} \quad (3)$$

where $||\cdot||_F$ is the Frobenius norm and $||\mathbf{x}||^2_\Omega = \mathbf{x}^\top \Omega \mathbf{x}$ for $x \in \mathbb{R}^2$ [2, 19].

Note that $\mathbf{R}_i \in \mathrm{SO}(2)$ for each $i \in \{1, \ldots, n\}$. $\mathrm{SO}(2)$ can be defined as follows:

$$\mathrm{SO}(2) = \left\{ \mathbf{R} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \in \mathbb{R}^{2\times 2} \;\middle|\; c^2 + s^2 = 1 \right\}. \quad (4)$$

This definition allows us to parameterize $\mathbf{H}_i$ and $\mathbf{H}_{ij}$ using (c_i, s_i, x_i, y_i) and $(c_{ij}, s_{ij}, x_{ij}, y_{ij})$ respectively,

$$\mathbf{H}_i = \begin{bmatrix} c_i & -s_i & x_i \\ s_i & c_i & y_i \end{bmatrix}, \mathbf{H}_{ij} = \begin{bmatrix} c_{ij} & -s_{ij} & x_{ij} \\ s_{ij} & c_{ij} & y_{ij} \end{bmatrix}, \quad (5)$$

as long as we enforce that $c_i^2 + s_i^2 = 1$. To simplify this notation, we define the sets $\mathbf{c} = \{c_1, \ldots, c_n\}$, $\mathbf{s} = \{s_1, \ldots, s_n\}$, $\mathbf{x} = \{x_1, \ldots, x_n\}$, and $\mathbf{y} = \{y_1, \ldots, y_n\}$.

If we evaluate the norms in (3) under this parameterization, then (3) is equivalent to

$$\begin{aligned} \underset{\mathbf{c},\mathbf{s},\mathbf{x},\mathbf{y}}{\operatorname{argmin}} \; & \sum_{(i,j)\in E} f_{ij}^{\mathbf{H}}(c_i, s_i, x_i, y_i, c_j, s_j, x_j, y_j) \quad (6) \\ \text{s.t.} \quad & c_i^2 + s_i^2 = 1, \quad \forall i \in \{1, \ldots, n\}, \end{aligned}$$

with,

$$\begin{aligned} f_{ij}^{\mathbf{H}}(c_i,s_i,x_i,y_i,c_j,s_j,x_j,y_j) &= f_{ij}^{Rot}(c_i,s_i,c_j,s_j) + \quad (7) \\ &+ f_{ij}^{Tran}(c_i,s_i,x_i,y_i,x_j,y_j), \end{aligned}$$

where

$$\begin{aligned} f_{ij}^{Rot}(c_i,s_i,c_j,s_j) &= \omega_{\mathbf{R}_{ij}}^2 (c_j - c_i c_{ij} + s_i s_{ij})^2 + \quad (8) \\ &+ \omega_{\mathbf{R}_{ij}}^2(-s_j + c_i s_{ij} + s_i c_{ij})^2 + \omega_{\mathbf{R}_{ij}}^2 (s_j - s_i c_{ij} + \quad (9) \\ &- c_i s_{ij})^2 + \omega_{\mathbf{R}_{ij}}^2 (c_j + s_i s_{ij} - c_i c_{ij})^2, \end{aligned}$$

and

$$\begin{aligned} f_{ij}^{Tran}(c_i,s_i,x_i,y_i,x_j,y_j) &= \omega_{x_{ij}}^2 (x_j - c_i x_{ij} + \quad (10) \\ &+ s_i y_{ij} - x_i)^2 + \omega_{y_{ij}}^2 (y_j - s_i x_{ij} - c_i y_{ij} - y_i)^2. \end{aligned}$$

Note that the cost is a polynomial in the space $\mathbb{R}[\mathbf{c}, \mathbf{s}, \mathbf{x}, \mathbf{y}]$ and that each individual term $f_{ij}^{\mathbf{H}} \in \mathbb{R}[c_i, s_i, x_i, y_i, c_j, s_j, x_j, y_j]$. We can also rewrite $c_i^2 + s_i^2 = 1$ as $0 \le 1 - c_i^2 - s_i^2 \le 1$ and $0 \le 2 - c_i^2 - s_i^2 \le 1$ for each $i \in \{1, \ldots, n\}$. This parameterization allows us to rewrite (6) as a polynomial optimization problem in the form described in (1) and (2), where $M = 2n$:

$$\begin{aligned} \underset{\mathbf{c},\mathbf{s},\mathbf{x},\mathbf{y}}{\operatorname{argmin}} \; & \sum_{(i,j)\in E} f_{ij}^{\mathbf{H}}(c_i, s_i, x_i, y_i, c_j, s_j, x_j, y_j) \quad (11) \\ \text{s.t.} \quad & 0 \le 1 - c_i^2 - s_i^2 \le 1, \; \forall i \in \{1, \ldots, n\}, \\ & 0 \le 2 - c_i^2 - s_i^2 \le 1, \; \forall i \in \{1, \ldots, n\}. \end{aligned}$$

C. Landmark SLAM

In Landmark SLAM, one estimates both the pose of the robotic vehicle at each time step, $(\mathbf{R}_i, \mathbf{t}_i) \in \mathrm{SE}(2)$, as well as the position of observed landmarks, $\mathbf{l}_\ell = [l_\ell^x, l_\ell^y]^\top \in \mathbb{R}^2$ for each $\ell \in \{1, \ldots, w\}$, given both relative pose measurements $(\bar{\mathbf{R}}_{ij}, \bar{\mathbf{t}}_{ij}) \in \mathrm{SE}(2)$ and landmark position observations $\bar{\mathbf{l}}_{i\ell} = [x_{i\ell}, y_{i\ell}]^\top \in \mathbb{R}^2$ that measure the position of landmark with respect to the local coordinate frame of the robot at the time step that it was observed. Let $L = \{i_k, \ell_k\}_{k=1}^{m_\ell} \subset \{1, \ldots, n\} \times \{1, \ldots, w\}$ identify the set of landmark position measurements where m_ℓ is the number of landmark measurements and let $\mathbf{l_x} = \{l_1^x, \ldots, l_w^x\}$ and $\mathbf{l_y} = \{l_1^y, \ldots, l_w^y\}$. We assume that the relative pose measurements are distributed according to the structure defined in the previous section and that $\bar{\mathbf{l}}_{i\ell} \sim \mathcal{N}(\mathbf{l}_{i\ell}^i, \Omega_{\mathbf{l}_{i\ell}}^{-1})$ where $\mathbf{l}_{i\ell}^i$ is the true position of the landmark ℓ with respect to the true pose $\mathbf{H}_i$ and $\Omega_{\mathbf{l}_{i\ell}} = \mathrm{blkdiag}(\omega_{x_{i\ell}}^2, \omega_{y_{i\ell}}^2)$ is the information matrix of $\bar{\mathbf{l}}_{i\ell}$.

Under these assumptions, the MLE solution to the Landmark SLAM problem can be written as follows:

$$\begin{aligned} \underset{\mathbf{c},\mathbf{s},\mathbf{x},\mathbf{y},\mathbf{l_x},\mathbf{l_y}}{\operatorname{argmin}} \; & \sum_{(i,j)\in E} f_{ij}^{\mathbf{H}}(c_i, s_i, x_i, y_i, c_j, s_j, x_j, y_j) + \quad (12) \\ & + \sum_{(i,\ell)\in L} f_{i\ell}^{Land}(c_i, s_i, x_i, y_i, l_\ell^x, l_\ell^y) \\ \text{s.t.} \quad & 0 \le 1 - c_i^2 - s_i^2 \le 1, \; \forall i \in \{1, \ldots, n\}, \\ & 0 \le 2 - c_i^2 - s_i^2 \le 1, \; \forall i \in \{1, \ldots, n\}. \end{aligned}$$

with,

$$\begin{aligned} f_{i\ell}^{Land}(c_i,s_i,x_i,y_i,l_\ell^x,l_\ell^y) &= \omega_{x_{i\ell}}^2 (l_\ell^x - c_i x_{i\ell} + \quad (13) \\ &+ s_i y_{i\ell} - x_i)^2 + \omega_{y_{i\ell}}^2 (l_\ell^y - s_i x_{i\ell} - c_i y_{i\ell} - y_i)^2. \end{aligned}$$

Note that the cost of the optimization problem in (12) is a polynomial in the space $\mathbb{R}[\mathbf{c}, \mathbf{s}, \mathbf{x}, \mathbf{y}, \mathbf{l_x}, \mathbf{l_y}]$ while $f_{i\ell}^{Land} \in \mathbb{R}[c_i, s_i, x_i, y_i, l_x, l_y]$. Also note that the constraints are the same as in (11) and thus, (12) is a polynomial optimization problem of the form defined in (1) and (2).

IV. Sparse Bounded Sum-of-Squares Programming

Polynomial optimization problems in general are non-convex, however, they can be approximated and sometimes solved exactly by solving a hierarchy of convex relaxations of the problem [20]. A variety of such convex relaxations hierarchies exist. This section covers a pair of such hierarchies. The first is called the bounded degree sum-of-squares (BSOS) hierarchy and consists of a sequence of SDP relaxations that can be used to find the globally optimal solution to small polynomial optimization problems that meet certain conditions [9]. The second is called Sparse-BSOS and enables us to leverage the sparsity inherent in SLAM problems to solve larger problem sizes than is possible using BSOS [10]. We conclude the section by describing the conditions that the cost and constraints that a polynomial optimization must satisfy for the first step of either hierarchy to converge exactly to the global optimum.

A. Bounded Sum-of-Squares

SOS programming is concerned with finding solutions to polynomial optimization problems as in (1). If $\mathbf{x}$ did not have to lie within the semi-algebraic set $\mathbf{K}$, solving the following problem would be equivalent to solving (1):

$$t^* = \sup_{t\in\mathbb{R}} \{t \mid f(\mathbf{x}) - t \ge 0, \forall\, \mathbf{x}\}. \quad (14)$$

If instead one had constraints g_j that bound the feasible space of the variable $\mathbf{x}$ to $\mathbf{K}$, then one would need to enforce that $f(\mathbf{x}) - t \ge 0$, $\forall\, \mathbf{x} \in \mathbf{K}$. At the same time, one would have to enforce it in a way that enabled $f - t$ to get as close to zero as possible at the optimal solution. Suppose we could optimize over a function h and also strictly enforce that it be non-negative on $\mathbf{K}$. Then, by enforcing that $f(\mathbf{x}) - t - h(\mathbf{x}) \ge 0$, for all x, we would equivalently enforce that $f(\mathbf{x}) - t \ge$

$h(\mathbf{x}) \geq 0$ on $\mathbf{K}$ and we would be able to optimize over h to minimize the gap between f and t on $\mathbf{K}$.

To apply this approach using numerical optimization, one would first need to know whether it was computationally tractable to enforce positivity of h on K. Assuming that $0 \leq g_j(x) \leq 1$ for all $x \in \mathbf{K}$ and $\mathbf{K}$ is compact, one can prove that if a polynomial h is strictly positive on $\mathbf{K}$, then h can be represented as

$$h(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{\alpha,\beta \in \mathbb{N}^M} \lambda_{\alpha\beta} \prod_j \left(g_j(\mathbf{x})^{\alpha_j} (1 - g_j(\mathbf{x}))^{\beta_j} \right), \quad (15)$$

for some (finitely many) nonnegative scalars $\boldsymbol{\lambda} = (\lambda_{\alpha\beta})$ [9, Theorem 1][1]. Conversely, any polynomial that can be written in the form defined in (15) is also positive on $\mathbf{K}$. This leads to a hierarchy of relaxations in which each relaxation bounds the number of monomial terms used to represent h [9, Theorem 2].

Let $N_d^{2M} = \{(\alpha, \beta) | \alpha, \beta \in \mathbb{N}^M, |\alpha| + |\beta| \leq d\}$ where the absolute value denotes the sum and

$$h_d(\mathbf{x}, \boldsymbol{\lambda}) := \sum_{(\alpha,\beta) \in N_d^{2M}} \lambda_{\alpha\beta} \prod_{j=1}^{M} g_j(\mathbf{x})^{\alpha_j} (1 - g_j(\mathbf{x}))^{\beta_j}, \quad (16)$$

where $\boldsymbol{\lambda} = (\lambda_{\alpha\beta}), (\alpha, \beta) \in \mathbb{N}_d^{2M}$. By choosing d, one can bound the number of monomial terms that are used to represent h_d and by optimizing over λ, one can optimize over the specific polynomial. By constraining $\boldsymbol{\lambda}$ to be non-negative, one can enforce that h_d be strictly positive on $\mathbf{K}$.

Now one can solve the following optimization problem:

$$t^* = \sup_{t,\boldsymbol{\lambda}} \{t | f(\mathbf{x}) - t - h_d(\mathbf{x}) \geq 0, \forall\ \mathbf{x}, \boldsymbol{\lambda} \geq 0\}. \quad (17)$$

However, optimizing over the space of all positive polynomials is computationally intractable. Instead, one can relax the problem again and optimize over the space of SOS polynomials up to a fixed degree since SOS polynomials are guaranteed to be positive and can be represented using a positive semidefinite matrix [20, Chapter 2]. Let $\Sigma[\mathbf{x}] \subset \mathbb{R}[\mathbf{x}]$ represent the space of SOS polynomials and let $\Sigma[\mathbf{x}]_k \subset \mathbb{R}[\mathbf{x}]_{2k}$ represent the space of SOS polynomials of degree at most $2k$. By fixing $k \in \mathbb{N}$, one arrives at the following BSOS family of convex relaxations: indexed by $d \in \mathbb{N}$:

$$q_d^k = \sup_{t,\boldsymbol{\lambda}} \{t | f(\mathbf{x}) - t - h_d(\mathbf{x}) \in \Sigma[\mathbf{x}]_k, \boldsymbol{\lambda} \geq 0\}. \quad (18)$$

Each of these optimization programs can be implemented as an SDP and provides a lower bound on the solution to (17). Additionally, it can be shown that under certain assumptions as $d \to \infty$, $q_d^k \to f^*$ [9, Theorem 2]. While this is useful for small problems, as the number of variables increases or for larger values of d and k, the runtime and memory usage of the optimization makes the use of this method infeasible [9, Section 3]. To address this challenge, we take advantage of sparsity in the optimization problem to dramatically scale problem size.

[1]Note that the theorem as presented requires the set $\{1, g_1, \ldots, g_M\}$ to generate $\mathbf{K}$, but since $\mathbf{K}$ is compact, one can always add a redundant linear constraint to the set to satisfy this requirement.

B. Sparse Bounded Sum-of-Squares

The Sparse-BSOS hierarchy takes advantage of the fact that for many optimization problems, the variables and constraints exhibit structured sparsity. It does this by splitting the variables in the problem into p blocks of variables and the cost into p associated terms, such that the number of variables and constraints relevant to each block is small [10].

Given $I \subset \{1, \ldots, N\}$, let $\mathbb{R}[\mathbf{x}; I]$ denote the ring of polynomials in the variables $\{x_i : i \in I\}$. Specifically Sparse-BSOS assumes that the cost and constraints satisfy the following assumption:

Assumption 1 (Running Intersection Property (RIP))**:** *There exists $p \in \mathbb{N}$ and $I_\ell \subseteq \{1, \ldots, N\}$ and $J_\ell \subseteq \{1, \ldots, M\}$ for all $\ell \in \{1, \ldots, p\}$ such that:*

- $f = \sum_{\ell=1}^{p} f^\ell$*, for some* $f^1, \ldots, f^p$*, such that* $f^\ell \in \mathbb{R}[\mathbf{x}, I_\ell]$ *for each* $\ell \in \{1, \ldots, p\}$*,*
- $g_j \in \mathbb{R}[\mathbf{x}, I_\ell]$ *for each* $j \in J_\ell$ *and* $\ell \in \{1, \ldots, p\}$*,*
- $\cup_{\ell=1}^{p} I_\ell = \{1, \ldots, N\}$*,*
- $\cup_{\ell=1}^{p} J_\ell = \{1, \ldots, M\}$*,*
- *for all* $\ell = 1, \ldots, p-1$*, there is an* $s \leq \ell$ *such that* $(I_{\ell+1} \cap \cup_{r=1}^{\ell} I_r) \subseteq I_s$*.*

In particular, I_ℓ denotes the variables that are relevant to ℓ-th block and J_ℓ denotes the associated relevant constraints. Intuitively, these blocks allow one to enforce positivity over a smaller set of variables which can reduce the computational burden while trying to solve this optimization problem.

We can use these definitions to define the Sparse-BSOS hierarchy that builds on the hierarchy defined in the previous section. Let $N^\ell := \{(\alpha, \beta) : \alpha, \beta \in \mathbb{N}_0, \text{supp}(\alpha) \cup \text{supp}(\beta) \subseteq J_\ell\}$, where $\mathbb{N}_0$ is the set of natural numbers including 0 and $\text{supp}(\alpha) := \{j \in \{1, \ldots, M\} : \alpha_j \neq 0\}$. Now let $N_d^\ell := \{(\alpha, \beta) \in N^\ell : \sum_j (\alpha_j + \beta_j) \leq d\}$, with $d \in \mathbb{N}$ and let

$$h_d^\ell(\mathbf{x}, \boldsymbol{\lambda}^\ell) := \sum_{(\alpha,\beta) \in N_d^\ell} \lambda_{\alpha\beta}^\ell \prod_{j=1}^{M} g_j(\mathbf{x})^{\alpha_j} (1 - g_j(\mathbf{x}))^{\beta_j}, \quad (19)$$

where $\boldsymbol{\lambda}^\ell \in \mathbb{R}^{|N_d^\ell|}$ is the vector of scalar coefficients $\lambda_{\alpha\beta}^\ell$. h_d^ℓ is again positive on $\mathbf{K}$ as long as the elements of $\boldsymbol{\lambda}$ are positive. If we again fix $k \in \mathbb{N}$, we can define a family of optimization problems indexed by $d \in \mathbb{N}$ as shown in (20), where $d_{\max}$ is defined on page 7 of [10].

This hierarchy of relaxations is called the Sparse-BSOS hierarchy and each level of the hierarchy can be implemented as an SDP. In addition, if RIP is satisfied, $0 \leq g_j(x) \leq 1$ for all $x \in \mathbf{K}$, and $\mathbf{K}$ is compact, then as $d \to \infty$, the sequence of optimization problems defined in (20) also converges to f^*[10, Theorem 2]. In addition a rank condition can be used to detect finite convergence [10, Lemma 4]. Importantly in particular cases, one can show that this optimization problem can be solved exactly when $d = 1$.

C. Sum-of-Squares Convexity

A polynomial $f \in \mathbb{R}[\mathbf{x}]$ is said to be SOS-convex the Hessian matrix $\mathbf{x} \mapsto \nabla^2 f(\mathbf{x})$ is an SOS matrix polynomial, that is, there exists an $a \in \mathbb{N}$ such that $\nabla^2 f = \mathbf{L}\mathbf{L}^\top$ for some real matrix polynomial $\mathbf{L} \in \mathbb{R}[\mathbf{x}]^{n \times a}$ for some a. A polynomial optimization problem of the form shown in

$$q_d^k = \sup_{\substack{t,\boldsymbol{\lambda}^1,\ldots,\boldsymbol{\lambda}^p,\\ f^1,\ldots,f^p}} \Big\{ t | f^\ell(\mathbf{x}) - h_d^\ell(\mathbf{x},\boldsymbol{\lambda}^\ell) \in \Sigma[\mathbf{x}; I_\ell]_k, \ell = 1,\ldots,p, \quad f(\mathbf{x}) - t = \sum_{\ell=1}^p f^\ell(\mathbf{x}), \boldsymbol{\lambda}^\ell \in \mathbb{R}^{|N_d^\ell|} \geq 0, t \in \mathbb{R}, f^\ell \in \mathbb{R}[\mathbf{x}; I_\ell]_{d_{max}} \Big\}. \tag{20}$$

(1) and (2) is said to be SOS-convex if the polynomials f^ℓ and $-g_j$ are SOS-convex for every $\ell \in \{1,\ldots,p\}$ and $j \in \{1,\ldots,M\}$. Importantly, when $k \in \mathbb{N}$ is fixed, the polynomials f^ℓ and g_j are at most degree $2k$ for each $\ell \in \{1,\ldots,p\}$ and $j \in \{1,\ldots,M\}$, and the optimization program is SOS-convex, then the Sparse-BSOS hierarchy defined in (20) converges at $d = 1$ [10, Theorem 3].

V. SBSOS-SLAM

We take advantage of the Sparse-BSOS relaxation hierarchy to solve the Pose Graph and Landmark SLAM problems. In this section, we prove that both these problems are SOS-convex as formulated in (11) and (12). Then, we talk about how we can enforce the RIP. Finally, we end with a discussion of implementation.

A. SOS-Convexity of Pose Graph and Landmark SLAM

To prove SOS-Convexity of the Pose Graph and Landmark SLAM problems, we use the following result:

Lemma 1: *[21, Lemma 2.2] A polynomial $f \in \mathbb{R}[\mathbf{x}]$ is SOS-convex if and only if the polynomial $\mathbf{z}^\intercal \nabla^2 f \mathbf{z}$ is a sum of squares in $\mathbb{R}[\mathbf{x}; \mathbf{z}]$.*

Using this result, one can show the following by noting the linearity of the Hessian:

Lemma 2: *If $f, g \in \mathbb{R}[\boldsymbol{x}]$ are SOS-convex, then $f + g$ is SOS-convex.*

We use these results to prove the following:

Theorem 3: *The Pose Graph SLAM problem defined in* (11) *can be solved exactly using the Sparse-BSOS optimization problem defined in* (20) *when $k = 1$ and $d = 1$.*

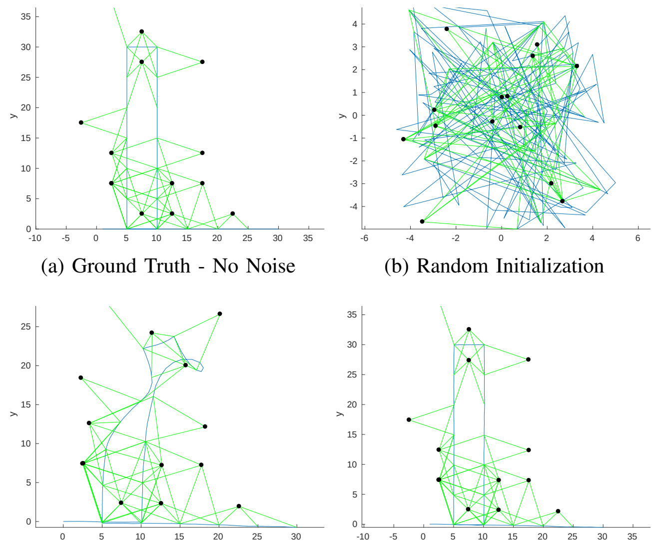
Proof. To prove this result, we show that the Pose Graph SLAM problem is SOS-convex and use [10, Theorem 3]. Note that $-g_i(c_i, s_i)$ can either be $c_i^2 + s_i^2 - 1$ or $c_i^2 + s_i^2 - 2$, which both produce the following Hessian:

$$\nabla^2(-g_i) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{2} \end{bmatrix}^\intercal \tag{21}$$

Therefore $-g_i$ is SOS-convex for all i. To show the cost function is SOS-convex, based on Lemma 2 it suffices to show $f_{ij}^{\mathbf{H}}(c_i, s_i, x_i, y_i, c_j, s_j, x_j, y_j)$ is SOS-convex, and it can be seen by looking at each of its elements.

- Let $\gamma_1(c_i, s_i, c_j) = \omega_{\mathbf{R}_{ij}}^2 (c_j - c_i c_{ij} + s_i s_{ij})^2$:

$$\nabla^2 \gamma_1 = \begin{bmatrix} \sqrt{2}\omega_{\mathbf{R}_{ij}} c_{ij} \\ -\sqrt{2}\omega_{\mathbf{R}_{ij}} s_{ij} \\ -\sqrt{2}\omega_{\mathbf{R}_{ij}} \end{bmatrix} \begin{bmatrix} \sqrt{2}\omega_{\mathbf{R}_{ij}} c_{ij} \\ -\sqrt{2}\omega_{\mathbf{R}_{ij}} s_{ij} \\ -\sqrt{2}\omega_{\mathbf{R}_{ij}} \end{bmatrix}^\intercal \tag{22}$$



(a) Ground Truth - No Noise (b) Random Initialization

(c) Levenberg-Marquardt (d) Global Optimal - SBSOS-SLAM

Fig. 2: Sample estimated Landmark SLAM solution for 100 nodes of CityTrees10000 dataset [8].

- Let $\gamma_2(c_i, s_i, s_j) = \omega_{\mathbf{R}_{ij}}^2 (-s_j + c_i s_{ij} + s_i c_{ij})^2$:

$$\nabla^2 \gamma_2 = \begin{bmatrix} \sqrt{2}\omega_{\mathbf{R}_{ij}} s_{ij} \\ \sqrt{2}\omega_{\mathbf{R}_{ij}} c_{ij} \\ -\sqrt{2}\omega_{\mathbf{R}_{ij}} \end{bmatrix} \begin{bmatrix} \sqrt{2}\omega_{\mathbf{R}_{ij}} s_{ij} \\ \sqrt{2}\omega_{\mathbf{R}_{ij}} c_{ij} \\ -\sqrt{2}\omega_{\mathbf{R}_{ij}} \end{bmatrix}^\intercal \tag{23}$$

- Let $\gamma_3(c_i, s_i, s_j) = \omega_{\mathbf{R}_{ij}}^2 (s_j - s_i c_{ij} - c_i s_{ij})^2$ and $\gamma_4(c_i, s_i, c_j) = \omega_{\mathbf{R}_{ij}}^2 (c_j + s_i s_{ij} - c_i c_{ij})^2$. Notice that $\gamma_3 = \gamma_2$ and $\gamma_4 = \gamma_1$, thus both γ_3 and γ_4 are also SOS-convex.
- Let $\gamma_5(c_i, s_i, x_i, x_j) = \omega_{x_{ij}}^2 (x_j - c_i x_{ij} + s_i y_{ij} - x_i)^2$:

$$\nabla^2 \gamma_5 = \begin{bmatrix} \sqrt{2}\omega_{x_{ij}} x_{ij} \\ -\sqrt{2}\omega_{x_{ij}} y_{ij} \\ \sqrt{2}\omega_{x_{ij}} \\ -\sqrt{2}\omega_{x_{ij}} \end{bmatrix} \begin{bmatrix} \sqrt{2}\omega_{x_{ij}} x_{ij} \\ -\sqrt{2}\omega_{x_{ij}} y_{ij} \\ \sqrt{2}\omega_{x_{ij}} \\ -\sqrt{2}\omega_{x_{ij}} \end{bmatrix}^\intercal \tag{24}$$

- Let $\gamma_6(c_i, s_i, y_i, y_j) = \omega_{y_{ij}}^2 (y_j - s_i x_{ij} - c_i y_{ij} - y_i)^2$:

$$\nabla^2 \gamma_6 = \begin{bmatrix} \sqrt{2}\omega_{y_{ij}} y_{ij} \\ \sqrt{2}\omega_{y_{ij}} x_{ij} \\ \sqrt{2}\omega_{y_{ij}} \\ -\sqrt{2}\omega_{y_{ij}} \end{bmatrix} \begin{bmatrix} \sqrt{2}\omega_{y_{ij}} y_{ij} \\ \sqrt{2}\omega_{y_{ij}} x_{ij} \\ \sqrt{2}\omega_{y_{ij}} \\ -\sqrt{2}\omega_{y_{ij}} \end{bmatrix}^\intercal \tag{25}$$

Since we can enlarge the domains of all γ_i's into $[c_i; s_i; c_j; s_j; x_i; x_j; y_i; y_j]$, and $f_{ij}^{\mathbf{H}} = \sum_{i=1}^6 \gamma_i$, we then conclude by Lemma 2 that $f_{ij}^{\mathbf{H}}$ is SOS-convex. □

By using a proof similar to Theorem 3, one can then show:

Theorem 4: *The Landmark SLAM problem defined in* (12) *can be solved exactly using the Sparse-BSOS optimization problem defined in* (20) *when* $k = 1$ *and* $d = 1$.

B. Satisfying the Running Intersection Property

The nature of the SLAM problem exhibits a large amount of sparsity [8, 15, 16]. However, to take advantage of the guarantees incumbent to the Sparse-BSOS hierarchy, we need to satisfy the RIP. An odometry chain forming the backbone of a SLAM graph inherently satisfies this property, however incorporating loop closures can make satisfying this assumption challenging. Finding the optimal selection of blocks I_ℓ is NP-hard, however a variety of heuristics exist. We used the algorithm defined in [22] to generate a sequence of variable groupings for the current implementation.

C. Implementation and Computational Scaling

For the experiments and development presented in this paper, we modified the code base released with [10] to formulate the problem as defined earlier on in the paper. In addition, we modified the code to convert the problem to a format where we could use the SDP solver within the optimization library Mosek [23].

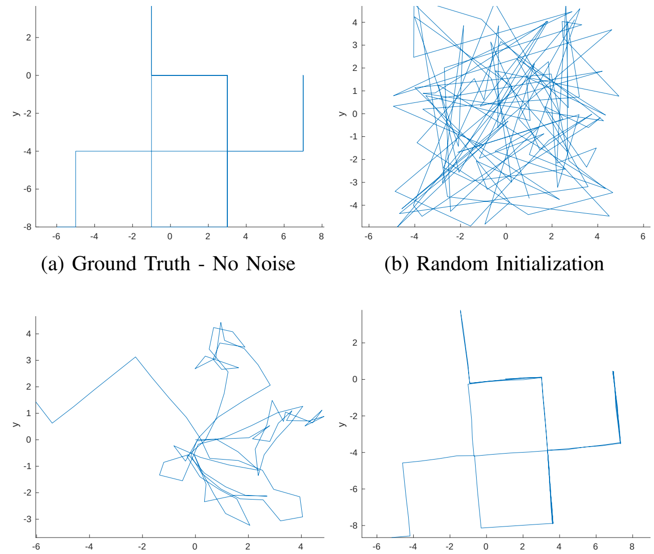
SOS programming optimizes over polynomials which can become ill-conditioned when optimization occurs over a large domain [10, Section 4]. Data can be scaled to address this problem, however if the optimization problem still remains poorly scaled then the optimization solver will warn the user that the problem cannot be satisfactorily solved.

VI. Experimental Proof of Concept

We evaluated the proposed convex SLAM algorithm on the CityTrees10000 [8] and Manhattan3500 [24] datasets by breaking the problem into sequences of 100 nodes and solving those graphs for which Mosek [23] did not run into numerical instabilities. For each dataset, we used the proposed SBSOS-SLAM methodology to find the globally optimal solution to the respective MLE problem defined in (11) and (12). For comparison, we also initialized Levenberg-Marquardt with a random initialization.

The median solve time for SBSOS-SLAM was 20.5507 seconds for the Manhattan3500 dataset and 161.8387 sec for the CityTree10000 dataset compared to less than a second on average for Levenberg-Marquardt. However, since our current implementation is based in Matlab and we are not attempting to satisfy the running intersection property optimally in these initial experiments, we believe there are a variety of extensions that can be made to improve scalability.

We show several example plots where Levenberg-Marquardt gets stuck in a local minima, while SBSOS-SLAM is guaranteed to find the global minimum and does not require initialization (Fig. 2, Fig. 3). Fig. 4 and Fig. 5 show that our proposed algorithm results in significantly smaller errors than Levenberg-Marquardt.



(a) Ground Truth - No Noise (b) Random Initialization

(c) Levenberg-Marquardt (d) Global Optimal - SBSOS-SLAM

Fig. 3: Sample estimated Pose Graph SLAM solution for 100 nodes of Manhattan3500 dataset [24].

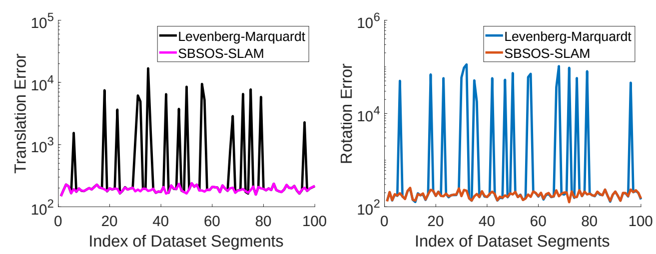


Fig. 4: Translational and rotational error verses groundtruth for the CityTrees10000 dataset.

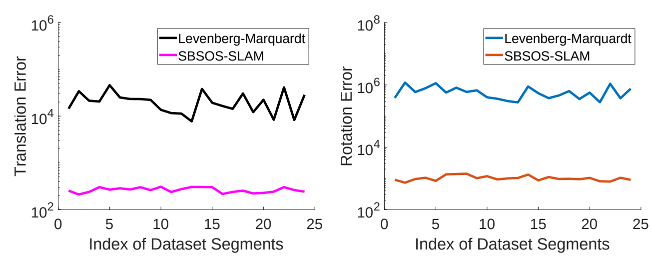


Fig. 5: Translational and rotational error verses groundtruth for the Manhattan3500 dataset.

VII. Conclusion

In this paper, we proposed an algorithm called SBSOS-SLAM that formulates the Pose Graph and Landmark SLAM problems as polynomial optimization programs that are sum-of-squares convex. As such, we are able to guarantee that we can find the optimal solution without any need for initialization.

References

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, present, and future of simultaneous

localization and mapping: Toward the robust-perception age," *IEEE Trans. on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] D. Rosen, L. Carlone, A. Bandeira, and J. Leonard, "SE-Sync: A certifiably correct algorithm for synchronization over the Special Euclidean group," in *Proc. Int. Work. Algorithmic Foundations of Robot.*, 2016.

[3] L. Carlone, G. C. Calafiore, C. Tommolillo, and F. Dellaert, "Planar pose graph optimization: Duality, optimal solutions, and verification." *IEEE Trans. on Robotics*, vol. 32, no. 3, pp. 545–565, 2016.

[4] D. M. Rosen, C. DuHadway, and J. J. Leonard, "A convex relaxation for approximate global optimization in simultaneous localization and mapping," in *Proc. IEEE Int. Conf. Robot. and Automation*, Seattle, Washington, USA, May 2015, pp. 5822–5829.

[5] L. Carlone, D. M. Rosen, G. Calafiore, J. J. Leonard, and F. Dellaert, "Lagrangian duality in 3D SLAM: Verification techniques and optimal solutions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, Hamburg, Germany, September 2015, pp. 125–132.

[6] G. Hu, K. Khosoussi, and S. Huang, "Towards a reliable slam back-end," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, Tokyo, Japan, Nov 2013, pp. 37–43.

[7] M. Liu, S. Huang, G. Dissanayake, and H. Wang, "A convex optimization based approach for pose slam problems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, Vilmoura, Portugal, Oct 2012, pp. 1898–1903.

[8] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[9] J. B. Lasserre, K.-C. Toh, and S. Yang, "A bounded degree SOS hierarchy for polynomial optimization," *EURO Journal on Computational Optimization*, vol. 5, no. 1-2, pp. 87–117, 2017.

[10] T. Weisser, J. B. Lasserre, and K.-C. Toh, "Sparse-BSOS: a bounded degree SOS hierarchy for large scale polynomial optimization with sparsity," *Mathematical Programming Computation*, vol. 10, no. 1, pp. 1–32, 2018.

[11] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping (SLAM): Part I," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, 2006.

[12] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 108–117, 2006.

[13] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[14] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Auton. Robot.*, vol. 4, no. 4, pp. 333–349, 1997.

[15] R. Eustice, M. Walter, and J. Leonard, "Sparse extended information filters: Insights into sparsification," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, Edmonton, AB, Canada, Aug 2005.

[16] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous localization and mapping via square root information smoothing," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1181–1203, 2006.

[17] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[18] N. Boumal, "A Riemannian low-rank method for optimization over semidefinite matrices with block-diagonal constraints," *ArXiv e-prints*, Jun. 2015.

[19] J. Briales and J. Gonzalez-Jimenez, "Cartan-Sync: Fast and global SE(d)-synchronization," *IEEE Robot. Autom. Letters*, vol. 2, 2017.

[20] J. B. Lasserre, *Moments, positive polynomials and their applications*. World Scientific, 2009, vol. 1.

[21] A. A. Ahmadi and P. A. Parrilo, "Sum of squares and polynomial convexity," in *Proc. IEEE Conf. Decision Control*, Shanghai, China, Dec 2009.

[22] L. Smail, "Junction trees constructions in bayesian networks," in *Journal of Physics: Conference Series*, vol. 893, no. 1. IOP Publishing, 2017, p. 012056.

[23] M. ApS, *The MOSEK C optimizer API manual Version 7.1 (Revision 54).*, 2015. [Online]. Available: http://docs.mosek.com/7.0/capi/

[24] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," in *Proc. IEEE Int. Conf. Robot. and Automation*, Orlando, Florida, May 2006, pp. 2262–2269.